Autonomous Space Rover

Giovanni Recchi (1849695)

January 2026

1 General Idea

The **Space Rover Exploration project** focuses on planning and decision-making for an autonomous rover exploring an unknown terrain. The rover must navigate different locations, avoid obstacles, collect scientific samples, and manage its energy levels efficiently.

To achieve this, we use **planning.domains** to define the problem and apply planners such as **LAMA** and **BFWF –FF-parsers version** to generate optimal action sequences.

2 Domain Description

The **Space Rover Exploration** domain models the operations of an autonomous rover navigating an extraterrestrial terrain. The rover's objective is to explore different locations, collect scientific samples, and manage its energy levels while avoiding obstacles.

2.1 Entities

- **Rover**: The autonomous vehicle exploring the terrain.
- **Location**: Different places in the environment where the rover can move.
- **Sample**: Scientific materials that the rover can collect.

2.2 Predicates (State Representations)

- `(at ?r - rover ?l - location)`: The rover is currently at a given location.
- `(connected ?l1 ?l2 - location)`: Two locations are directly connected, allowing movement between them.
- `(energy-empty ?r - rover)`: The rover has no energy left.
- `(energy-low ?r - rover)`, `(energy-medium ?r - rover)`, `(energy-high ?r - rover)`: Different energy levels of the rover.
- `(has-sample ?r - rover ?s - sample)`: The rover has collected a particular sample.
- `(sample-at ?s - sample ?l - location)`: A sample is located at a specific location.
- `(obstacle ?l - location)`: There is an obstacle at a given location that prevents movement.
- `(sunny ?l - location)`: A location where the rover can recharge using solar panels.

2.3 Actions

2.3.1 Move

- The rover can move between connected locations if there are no obstacles.
- Movement is only possible when the rover has at least some energy.
- The rover's energy level decrease after moving.
- The new location is marked as visited.

2.3.2 Recharge

- If the rover is in a sunny location, it can use solar energy to recharge its battery.

2.3.3 Collect Sample

- The rover can collect a sample from a location where a sample is available.
- Collection is only possible when the rover has at least some energy.
- The rover energy level decrease after collecting.

2.3.4 Destroy Obstacles

- The rover can destroy an obstacle with a laser in a location connected to the location where it is at.
- The shot is only possible when the rover has at least some energy.
- The rover energy level decrease after shot.

3 Instances

We create three different instances that the PDDL planner can solve. In general, the first instance is the simplest, while the last is the most difficult to solve.

3.1 Instance 1

Instance 1 models a simple scenario in which a rover must navigate a predefined terrain while managing its energy levels. The objective is to move the rover from its starting position (`loc1`) to a designated goal location (`loc4`) while considering constraints such as connectivity and energy consumption.

The **initial state** is the following: the rover starts at `loc1` with high energy. The terrain consists of connected locations: `loc1` to `loc2`, `loc2` to `loc3`, `loc3` to `loc4`, and `loc4` to `loc5`. A sample is placed at `loc2`, and `loc3` is a sunny location where the rover can recharge. The **goal** of this instance is for the rover to reach `loc4`, so the rover will ignore the sample at `loc2`.

The rover must move between connected locations while managing its energy levels. If necessary, it may need to recharge at `loc3`. Collecting the sample at `loc2` is an optional task.

3.2 Instance 2

In this scenario, the rover must navigate a terrain, collect a sample, and reach a specific goal location while avoiding obstacles and managing energy efficiently. The challenge lies in finding an alternative path or removing an obstruction.

The **initial state** is the following: the rover starts at `loc1` with a high energy level. Locations are connected sequentially from `loc1` to `loc5`, but an obstacle at `loc4` blocks direct passage. However, the rover has the ability to destroy obstacles when it is in a location connected to the obstacle's location. The sample is located at `loc5`, and both `loc3` and `loc5` are sunny locations where the rover can recharge. The rover must collect the sample at `loc5` and successfully reach `loc3`.

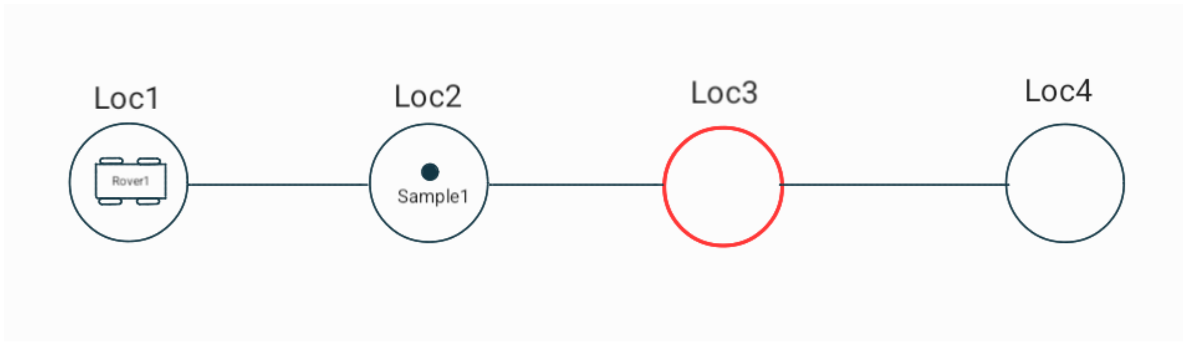

Figure 1: This is the environment of the instance 1 (red circle=location where the rover can recharge using solar panels).

The rover must remove the obstacle at `loc4`. The rover must ensure the obstacle has sufficient energy for the action. Efficient energy management is crucial, and using sunny locations for recharging may be necessary for completing the mission successfully.

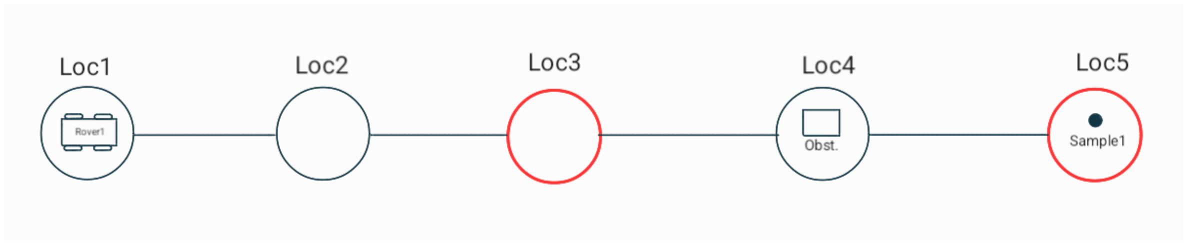

Figure 2: This is the environment of the instance 2.

3.3 Instance 3

In this scenario, two rovers must work together to collect multiple samples, manage their energy levels, and return to their respective starting locations. The challenge includes navigating around or removing obstacles and recharging when necessary.

The **initial state** is the following: `rover1` starts at `loc1`, and `rover2` starts at `loc6`, both with high energy levels. Locations are interconnected, forming a navigable terrain. However, obstacles are present at `loc2`, `loc4`, and `loc5`. The rovers have the ability to destroy obstacles when in a location connected to the obstacle's location. The samples are distributed across the environment: `sample1` at `loc4`, `sample2` at `loc7`, and `sample3` at `loc8`. Sunny locations where the rovers can recharge are `loc3`, `loc7`, and `loc8`. `rover1` must collect `sample1` and `sample2` and return to `loc1`, while `rover2` must collect `sample3` and return to `loc6`.

The rovers must navigate around or remove obstacles at `loc2`, `loc4`, and `loc5`. Efficient energy management is essential, requiring the use of sunny locations for recharging. The coordination between the two rovers is crucial to complete the task successfully.

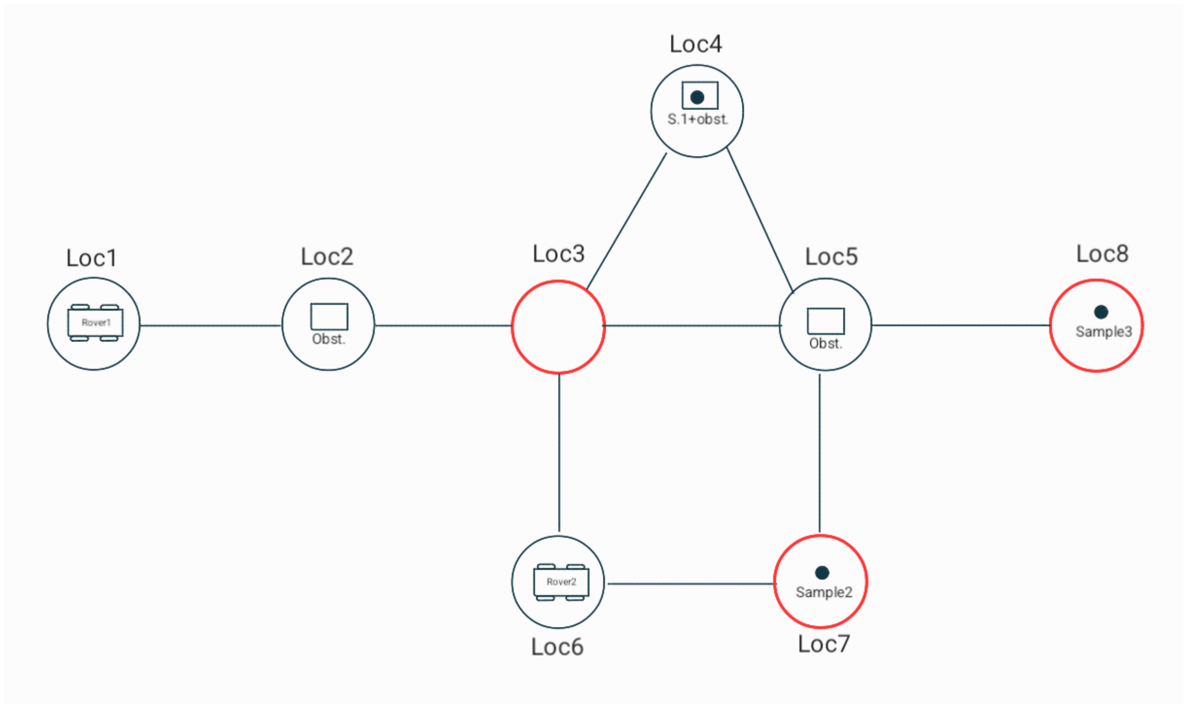

Figure 3: This is the environment of the instance 3.

4 Planning Environment

4.1 Workflow

All planning tasks are modeled in **PDDL** through one domain file and three problem instances. We solved each instance using the `planning.domains` platform, running two different planners:

- **LAMA** (Landmark-based planner);
- **BFWS–FF Parser Version** (width-based search with FF heuristic guidance).

For each run, we saved the plan returned by the planner as a sequence of grounded actions. In our saved outputs, the available information is the **plan itself** (i.e., the action list), therefore the evaluation focuses on:

- **Plan length**: number of actions in the solution;

- **Action-type distribution**: number of moves, recharges, obstacle destructions, and sample collections.

4.2 LAMA

LAMA is a heuristic planner that exploits **landmarks** (subgoals that must be achieved in any valid plan) and combines greedy search with refinement phases (e.g., weighted A*). In our domain, landmarks are naturally related to reaching key locations, removing blocking obstacles when necessary, and ensuring sample collection before returning to the goal locations.

4.3 BFWS–FF Parser Version

BFWS (Best-First Width Search) prioritizes states based on **novelty** (new combinations of relevant facts) and uses the **FF heuristic** to guide exploration. This can be beneficial in domains where the planner must balance exploration (reaching distant locations) with constraints (obstacles and energy management).

4.4 Results and Metrics

Table 1 summarizes the results extracted from the generated plans. For Instances 1 and 2, the plan found in our runs is the same (hence we report a single plan per instance). For Instance 3, we report both solutions (LAMA and BFWS), which differ slightly in length and structure.

Instance	**Planner**	**Plan length**	**#move**	**#recharge**	**#destroy / #collect**
1	LAMA / BFWS	5	3	1	0 / 1
2	LAMA / BFWS	13	8	3	1 / 1
3	LAMA	29	17	6	3 / 3
3	BFWS–FF	28	16	6	3 / 3

Table 1: Planning results extracted from the generated plans: plan length and action-type distribution.

4.5 Discussion

A few qualitative observations emerge from the produced plans:

- **Instance 1 (5 actions).** The plan performs a short navigation, collects the required sample, recharges at the sunny location, and reaches the goal location.
- **Instance 2 (13 actions).** The plan includes obstacle removal, then navigation to the sample location, sample collection, and finally returning to the goal location, with intermediate recharges to maintain feasibility.
- **Instance 3 (multi-rover).** Both planners solve the problem by distributing tasks across the two rovers and using recharges to ensure feasibility. BFWS produces a slightly shorter plan (28 vs 29 actions) mainly by reducing one movement step, while keeping the same number of recharges and required task actions (destroy/collect).

These results show how increasing instance complexity increases plan length and requires more frequent energy management (recharge actions) and obstacle handling.

5 Conclusion

In this project we modeled an **autonomous space rover exploration** scenario as a classical planning problem in **PDDL**. The domain captures navigation on a graph of locations, the presence of obstacles, the possibility to recharge energy in sunny locations, and the need to collect scientific samples.

We designed three instances with increasing complexity. The obtained plans reflect a clear growth in difficulty: Instance 1 requires only a short navigation with minimal energy management, Instance 2 introduces obstacle removal and a longer route that forces additional recharges, and Instance 3 extends the setting to a **multi-rover** scenario with multiple samples and obstacles, increasing both the length and the structure of the plan.

We tested two heuristic planners (LAMA and BFWS–FF). The solutions show that both approaches can solve the considered instances, while producing slightly different plans in the most complex case. In particular, for Instance 3 the two planners generated plans with comparable structure (same number of required task actions such as sample collection and obstacle removal), with small differences in the navigation part, which resulted in a minor difference in plan length. Overall, the experiments confirm that increasing environmental complexity leads to longer plans and a higher number of energy-management and obstacle-handling actions, highlighting the relevance of heuristics and exploration strategies in this domain.

6 Future Work

Several extensions could improve both the realism of the model and the experimental analysis:

- **Richer cost model.** Introduce action costs (e.g., different movement costs, laser usage costs, or energy consumption per action) and optimize for total cost rather than plan length.
- **Stronger evaluation.** Collect additional metrics from the planner output (e.g., expanded/-generated states and runtime under controlled hardware conditions) and run multiple trials to reduce variability.
- **Larger benchmark set.** Create additional instances with increasing numbers of locations, obstacles, and samples to better evaluate scalability and identify the breaking point of different planners.
- **Coordination constraints.** Extend the multi-rover instance with shared resources (e.g., a single charging station, limited communication, or mutual exclusion constraints) to stress cooperation and scheduling aspects.
- **Uncertainty.** Model exogenous events (e.g., new obstacles appearing, solar storms, or temporary blocked connections) using contingent planning or replanning approaches.